FORSCHUNGSZENTRUM JÜLICH GmbH
Zentralinstitut für Angewandte Mathematik
D-52425 Jülich, Tel. (02461) 61-6402

Interner Bericht

Computer-aided Modelling and Simulation of the Thermodynamics of Steam Reforming

*Johannes Grotendorst, Jürgen Dornseiffer**

FZJ-ZAM-IB-9812

August 1998

(letzte Änderung: 18.08.98)

(*) Department of Chemistry and Dynamics of the Geosphere
Institute for Applied Physical Chemistry

Preprint: Mathematics and Computers in Simulation 1738 (1999) 1-21

Computer-aided Modelling and Simulation of the Thermodynamics of Steam Reforming

J. Grotendorst [a,1] and J. Dornseiffer [b]

[a] *Central Institute for Applied Mathematics*

[b] *Department of Chemistry and Dynamics of the Geosphere*
Institute for Applied Physical Chemistry

Forschungszentrum Jülich
D-52425 Jülich
Germany

Abstract

In this article we analyse the thermodynamics of steam reforming by using computing techniques from symbolics, numerics and graphics. It is shown that the integration of the various computing methods normally used separately in scientific problem-solving allows the development of efficient and flexible tools for computer-aided modelling and simulation. We calculate the product gas composition at equilibrium by solving a multiparameter nonlinear equation system, which itself is derived by reformulating the molecular mass balance equations. The effects of the operating parameters of temperature, pressure and the steam to carbon ratio on the equilibrium gas composition are studied below and above the carbon boundary. The thermodynamic model was used to determine the mass and energy balance of an experimental steam reformer for converting organic wastes (fluids and gases) into hydrogen-rich product gases.

1 General considerations

The concept of scientific computing has evolved from a field encompassing primarily numerical methods to a much broader field that includes algebraic and analytical methods, numerical methods, and graphics. The combination of all these computing techniques facilitates efficient and accurate problem-solving. It allows the development of software systems for the automatic solution of problems in ways not possible with conventional computing systems.

[1] Corresponding author. E-mail: j.grotendorst@fz-juelich.de

2 September 1998

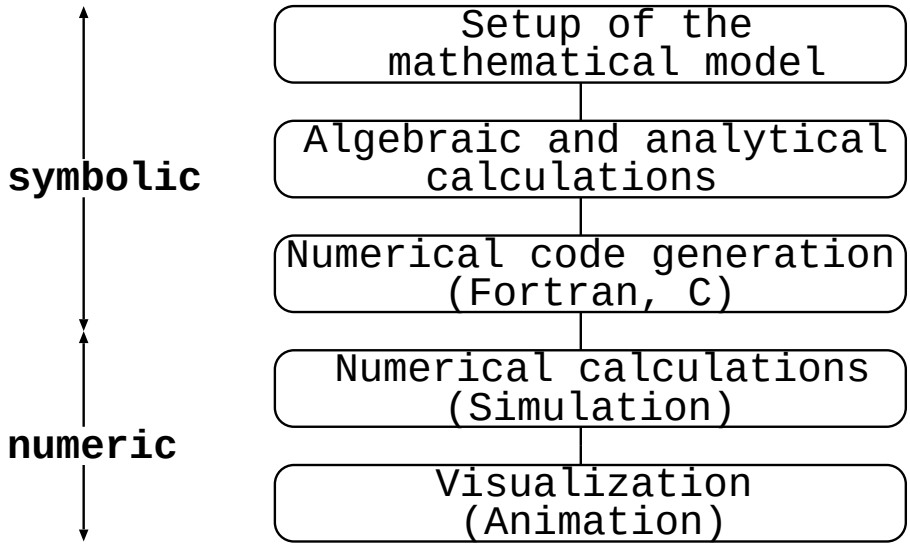

Fig. 1. Computing steps in a scientific problem-solving process.

Fig. 1 displays typical steps in a computerized solution process. In previous articles in this journal [1] Ross Taylor discussed the advantages of using an integrated software system for symbolic manipulations, numerical computations and graphical visualization like Maple [2] in thermodynamics. In this article we extend this approach and give an illustrative and detailed example from thermodynamics demonstrating a general concept for computer-aided modelling and simulation in science and engineering.

2 Thermodynamic model

The catalytic reaction between steam and hydrocarbons into mixtures of hydrogen, carbon monoxide, carbon dioxide and methane forms the basic industrial feedstock (synthesis gas) to produce ammonia, methanol and other chemicals. The theoretical study of the reaction behaviour is important for reactor design, e.g. to determine the mass and energy balance of a steam reformer, and for finding the most economic reaction conditions, especially the carbon formation regions. We investigate the dependence of the operating parameters of temperature, pressure and the steam to carbon ratio on the equilibrium gas composition below and above the carbon boundaries.

Steam reforming of a hydrocarbon can be described by the following simple system of reaction equations, where the carbon formation reaction (3) is only present under special reaction conditions (below the carbon boundary):

1. The methane-steam equality

$$CH_4 + H_2O \rightleftharpoons CO + 3H_2$$

2. The water-gas shift

$$CO + H_2O \rightleftharpoons CO_2 + H_2$$

3. The carbon-methanation equality

$$CH_4 \rightleftharpoons C + 2H_2$$

To set up a practical computational model we represent the equilibrium composition of the product gas in terms of the reaction extents ξ_i, $i = 1..3$, for each separate reaction. If ne_j, $j = 1..5$, is the number of moles of chemical species S_j present at equilibrium and n_j is the initial number of moles of that species, then we have

$$ne_j = n_j + \sum_{i=1}^{3} \alpha_{ji}\xi_i,$$

where α_{ji} is the stoichiometric coefficient of species S_j in each of the reaction equations (i) [3]. The following table comprises the matrix of stoichiometric coefficients and the product gas components at equilibrium in terms of the reaction extents.

S_j	$\alpha_{j,1}$	$\alpha_{j,2}$	$\alpha_{j,3}$	ne_j
CH_4	-1	0	-1	$n_1 - \xi_1 - \xi_3$
H_2O	-1	-1	0	$n_2 - \xi_1 - \xi_2$
CO	1	-1	0	$n_3 + \xi_1 - \xi_2$
H_2	3	1	2	$n_4 + 3\xi_1 + \xi_2 + 2\xi_3$
CO_2	0	1	0	$n_5 + \xi_2$
				$\sum ne_j = \sum n_j + 2\xi_1 + \xi_3$

3 Generation of the mathematical model

In this section we describe the derivation of the mathematical model. The Maple code is included to illustrate the strategy used (Maple commands will occur on lines beginning with ">"). First we load the linear algebra package linalg whose functions will be used quite frequently in the following.

```
> with(linalg):
```

For creating the matrix of stoichiometric coefficients we define a short Maple function.

```
> stoich:=(eq::list(equation),l::list) ->
  concat(seq(map(x->coeff((rhs-lhs)(eq[i]),x),l),i=1..nops(eq))):
```

The function stoich generates the stoichiometric matrix from a list of reaction equations and a list of participating chemical components; the linalg function concat joins two or more column vectors together horizontally. In the procedure stoich we applied the new language feature lexical scoping in Maple V Release 5 [2] (lexical scoping allows a nested procedure to access the variables which are located in surrounding procedures). For the three steam reforming reactions we get

```
> eqs:=[CH4+H2O=CO+3*H2,CO+H2O=CO2+H2,CH4=C+2*H2]:
  che:=[CH4,H2O,CO,H2,CO2,C]:
  alpha:=stoich(eqs,che[1..5]);
```

$$\alpha := \begin{bmatrix} -1 & 0 & -1 \\ -1 & -1 & 0 \\ 1 & -1 & 0 \\ 3 & 1 & 2 \\ 0 & 1 & 0 \end{bmatrix}$$

where the rank of this matrix

```
> rank(alpha);
```

3

shows that these equations are linearly independent. The gas composition at equilibrium is obtained by simple matrix algebra:

```
> xi:=vector(3): n:=vector(5):
  ne:=evalm(n + alpha &* xi): print(convert(ne,matrix));
```

$$\begin{bmatrix} n_1 - \xi_1 - \xi_3 \\ n_2 - \xi_1 - \xi_2 \\ n_3 + \xi_1 - \xi_2 \\ n_4 + 3\,\xi_1 + \xi_2 + 2\,\xi_3 \\ n_5 + \xi_2 \end{bmatrix}$$

Here `&*` denotes the operator for non-commutative matrix multiplication in Maple. The resulting row vector `ne` is converted to a column vector for printing purposes and the initial gas quantities are indicated by

```
> seq(n[i]=n[che[i]],i=1..5);
```

$$n_1 = n_{CH4},\ n_2 = n_{H2O},\ n_3 = n_{CO},\ n_4 = n_{H2},\ n_5 = n_{CO2}$$

Hence, a feed mixture of n_1 moles of methane and n_2 moles of steam forms the following gas components:

```
> n[3]:=0: n[4]:=0: n[5]:=0:
  convert(evalm(n + alpha &* xi),matrix);
```

$$\begin{bmatrix} n_1 - \xi_1 - \xi_3 \\ n_2 - \xi_1 - \xi_2 \\ \xi_1 - \xi_2 \\ 3\,\xi_1 + \xi_2 + 2\,\xi_3 \\ \xi_2 \end{bmatrix}$$

For calculating the gas composition of a general hydrocarbon feedstock, e.g. hexane, the same three reaction equations (1-3) are applicable because methane is thermodynamically the most stable hydrocarbon and one of the reaction products in steam reforming. In comparison to methane reforming with a hydrogen-carbon ratio of $\eta = 4$, we have to consider a hydrogen deficiency of $(4-\eta)/2$ moles per atom of carbon in the feedstock (i.e. for hexane C_6H_{14} we have $(4-\eta)/2 = 5/6$ with $\eta = 14/6$). Thus, for a general hydrocarbon feedstock with n_1 moles of carbon and n_2 moles of steam we obtain the following vector with gas components:

```
> n[4]:=-(2-eta/2)*n[1]:
  ne:=evalm(n + alpha &* xi): print(convert(ne,matrix));
```

$$\begin{bmatrix} n_1 - \xi_1 - \xi_3 \\ n_2 - \xi_1 - \xi_2 \\ \xi_1 - \xi_2 \\ -(2 - \frac{1}{2}\,\eta)\, n_1 + 3\,\xi_1 + \xi_2 + 2\,\xi_3 \\ \\ \xi_2 \end{bmatrix}$$

The quantities ne_j (number of moles of chemical species S_j in the product gas) simultaneously satisfy the equilibrium conditions for each separate reaction (i). For gas reactions the mass balance equations are defined in terms of partial pressures.

```
> eqn[i]:=K[i]= Product(p[j]^alpha[j,i],j=1..5):
  print(%);
```

$$K_i = \prod_{j=1}^{5} {p_j}^{\alpha_{j,\,i}}$$

Here K_i denotes the equilibrium constant of the reaction equation (i) and p_j the partial pressure of gas component S_j.

```
> p:=vector(5,i->P[che[i]]);
```

$$p := [P_{CH4},\, P_{H2O},\, P_{CO},\, P_{H2},\, P_{CO2}]$$

These equations have the following explicit form:

```
> for i from 1 to 3 do
     print(value(eqn['i']))
  od:
```

$$K_1 = \frac{P_{CO}\, {P_{H2}}^3}{P_{CH4}\, P_{H2O}}$$

$$K_2 = \frac{P_{H2}\, P_{CO2}}{P_{H2O}\, P_{CO}}$$

$$K_3 = \frac{{P_{H2}}^2}{P_{CH4}}$$

Next, we insert the relation $p_j = x_j P$, where x_j denotes the mole fraction of gas component S_j and P the total pressure.

```
> p:=seq(x[j]*P, j=1..5):
  for i from 1 to 3 do
     print(value(eqn['i']));
  od:
```

$$K_1 = \frac{P^2\, x_3\, {x_4}^3}{x_1\, x_2}$$

$$K_2 = \frac{x_4\, x_5}{x_2\, x_3}$$

$$K_3 = \frac{P\, {x_4}^2}{x_1}$$

The mole fraction x_j itself is defined by dividing ne_j by the total number of moles, i.e. the sum over all components should yield unity.

```
> nsum:=sum(ne[j],j=1..5):
  x:=map(y -> y/nsum,ne);
```

$$x := \left[\frac{n_1 - \xi_1 - \xi_3}{\%2},\ \frac{n_2 - \xi_1 - \xi_2}{\%2},\ \frac{\xi_1 - \xi_2}{\%2},\ \frac{-\%1 + 3\,\xi_1 + \xi_2 + 2\,\xi_3}{\%2},\ \frac{\xi_2}{\%2}\right]$$

$$\%1 := (2 - \frac{1}{2}\,\eta)\, n_1$$

$$\%2 := n_1 + 2\,\xi_1 + \xi_3 + n_2 - \%1$$

```
> Sum(x[j],j=1..5)=simplify(sum(x[j],j=1..5));
```

$$\sum_{j=1}^{5} x_j = 1$$

To complete the computational model we substitute the mole factions x_j, $j = 1..3$, into the mass balance equations and get the following three multiparameter equations:

```
> for i from 1 to 3 do
     f[i]:=ln(value(rhs(eqn['i'])))-ln(lhs(eqn['i']))=0:
     print(%);
  od:
```

$$\ln\left(\frac{P^2\,(\xi_1-\xi_2)\,(-(2-\frac{1}{2}\,\eta)\,n_1+3\,\xi_1+\xi_2+2\,\xi_3)^3}{(n_1-\xi_1-\xi_3)\,(n_1+2\,\xi_1+\xi_3+n_2-(2-\frac{1}{2}\,\eta)\,n_1)^2\,(n_2-\xi_1-\xi_2)}\right)$$

$$-\ln(K_1)=0$$

$$\ln\left(\frac{(-(2-\frac{1}{2}\,\eta)\,n_1+3\,\xi_1+\xi_2+2\,\xi_3)\,\xi_2}{(n_2-\xi_1-\xi_2)\,(\xi_1-\xi_2)}\right)-\ln(K_2)=0$$

$$\ln\left(\frac{P\,(-(2-\frac{1}{2}\,\eta)\,n_1+3\,\xi_1+\xi_2+2\,\xi_3)^2}{(n_1-\xi_1-\xi_3)\,(n_1+2\,\xi_1+\xi_3+n_2-(2-\frac{1}{2}\,\eta)\,n_1)}\right)-\ln(K_3)=0$$

For analysing the thermodynamics of steam reforming we have to solve these nonlinear equations for the reaction extents ξ_i, $i=1..3$, simultaneously.

4 Simple numerical calculations

First we study the steam reforming of hexane at $T=1000\,K$ under different reaction conditions. The equilibrium constants required at this temperature can be calculated by using the corresponding values for the pure substances [4]. For K_i, $i=1..3$, we get

```
> K[1]:=26.122: K[2]:=1.406: K[3]:=10.519:
```

For a feed mixture of $n_1=1$ mole of carbon and $n_2=1$ mole of steam under a pressure of $P=2\,bar$ and with a hydrogen-carbon ratio of $\eta=14/6$ we calculate the reaction extents by solving the equation system numerically with the Maple procedure fsolve. Assuming an accuracy of 4 digits we obtain

```
> n[1]:=1: n[2]:=1: P:=2: eta:=14/6: Digits:=4:
  sol:=fsolve({f[1],f[2],f[3]},{xi[1]=0.5..0.7,xi[2]=0..0.2,
  xi[3]=0..0.2}): print(op(sol));
```

$$\xi_1=.6350,\ \xi_2=.1161,\ \xi_3=.1878$$

and for the corresponding mole fractions:

```
> s:=map(y -> eval(y,sol),x):
  seq(x[che[i]]=s[i],i=1..5);
```

$$x_{CH4} = .06750,\ x_{H2O} = .09482,\ x_{CO} = .1977,\ x_{H2} = .5958,\ x_{CO2} = .04423$$

The amount of carbon deposition is given by

```
> n[C]=eval(xi[3],sol)*n[1]*mole;
```

$$n_C = .1878\, mole$$

Therefore, under these reaction conditions we have strong carbon formation in the reactor. Increasing the steam to carbon ratio $N = n_2/n_1$ from 1 to 1.5 we obtain:

```
> n[2]:=1.5:
  fsolve({f[1],f[2],f[3]},{xi[1]=0.7..1,xi[2]=0..0.3,
  xi[3]=-0.2..0}): print(op(%));
```

$$\xi_1 = .9607,\ \xi_2 = .1942,\ \xi_3 = -.1643$$

Now, the reaction system is above the carbon boundary (negative ξ_3 means no carbon formation). In this situation the carbon-methanation reaction (3) is not needed for the evaluation of the product gas composition at equilibrium, i.e. we have to solve the first two reaction equations with $\xi_3 = 0$:

```
> fsolve({eval(f[1],xi[3]=0),eval(f[2],xi[3]=0)},
  {xi[1]=0.7..1,xi[2]=0..0.3}): print(op(%));
```

$$\xi_1 = .8552,\ \xi_2 = .2063$$

The values for the reaction extents are slightly different when solving the two-dimensional equation system for ξ_1 and ξ_2.

```
> P:='P': eta:='eta': n[1]:='n[1]': n[2]:='n[2]':
  K[1]:='K[1]': K[2]:='K[2]': K[3]:='K[3]': Digits:=10:
  eval(f[1],xi[3]=0); eval(f[2],xi[3]=0);
```

$$\ln\left(\frac{P^2\,(\xi_1-\xi_2)\,(-(2-\frac{1}{2}\,\eta)\,n_1+3\,\xi_1+\xi_2)^3}{(n_1-\xi_1)\,(n_1+2\,\xi_1+n_2-(2-\frac{1}{2}\,\eta)\,n_1)^2\,(n_2-\xi_1-\xi_2)}\right)-\ln(K_1)=0$$

$$\ln\left(\frac{(-(2-\frac{1}{2}\,\eta)\,n_1+3\,\xi_1+\xi_2)\,\xi_2}{(n_2-\xi_1-\xi_2)\,(\xi_1-\xi_2)}\right)-\ln(K_2)=0$$

5 Modelling a temperature-dependent expression for K_i

To simplify the subsequent parameter studies we replace the equilibrium constants K_i by analytical expressions in T. First we derive an expression for the reaction enthalpy by using Kirchhoff's law in the form

$$\frac{\partial}{\partial T} H_P(T) = \Delta c_p$$

with a polynomial ansatz for the molar heat capacity [4].

$$\Delta c_p = \sum_{j=0}^{6} A_j T^j$$

In Maple we solve the following simple differential equation:

```
> Diff(H[P](T),T)=Sum(A[j]*(s*T)^j,j=0..6);
```

$$\frac{\partial}{\partial T} H_P(T) = \sum_{j=0}^{6} A_j \, (s\,T)^j$$

```
> dsolve(value(%),H[P](T));
  assign(%): HPT:=rhs(%):
```

$$H_P(T) = A_0\,T + \frac{1}{2}\,A_1\,s\,T^2 + \frac{1}{3}\,A_2\,s^2\,T^3 + \frac{1}{4}\,A_3\,s^3\,T^4 + \frac{1}{5}\,A_4\,s^4\,T^5 + \frac{1}{6}\,A_5\,s^5\,T^6 + \frac{1}{7}\,A_6\,s^6\,T^7 + \_C1$$

Here A_j, $j = 0..6$, denote the coefficients of the polynomial and $s(= 10^{-3})$ defines a scaling factor of the temperature T. Inserting the expression for $H_P(T)$ into the equation of van't Hoff and then integrating yields:

```
> Diff(lnK[P](T),T)='H[P](T)'/(R*T^2);
```

$$\frac{\partial}{\partial T} lnK_P(T) = \frac{H_P(T)}{R\,T^2}$$

```
> dsolve(%,lnK[P](T));
  lnKPT:=rhs(%):
```

$$lnK_P(T) = \frac{1}{42}\,\frac{A_6\,s^6\,T^6}{R} + \frac{1}{30}\,\frac{A_5\,s^5\,T^5}{R} + \frac{1}{20}\,\frac{A_4\,s^4\,T^4}{R} + \frac{1}{12}\,\frac{A_3\,s^3\,T^3}{R} +$$

$$\frac{1}{6}\frac{A_2\,s^2\,T^2}{R}+\frac{1}{2}\frac{A_1\,s\,T}{R}-\frac{\_C1}{R\,T}+\frac{A_0\ln(T)}{R}+\_C2$$

The coefficients A_j are determined by the corresponding coefficients of the pure substances in each reaction. These coefficients are obtainable from thermochemical data prepared for computer calculations [4]. The following coefficient matrix was used:

```
> U:=matrix(7,6,(i,j) -> a.(i-1)[che[j]]);
```

$$U := \begin{bmatrix} a0_{CH4} & a0_{H2O} & a0_{CO} & a0_{H2} & a0_{CO2} & a0_{C} \\ a1_{CH4} & a1_{H2O} & a1_{CO} & a1_{H2} & a1_{CO2} & a1_{C} \\ a2_{CH4} & a2_{H2O} & a2_{CO} & a2_{H2} & a2_{CO2} & a2_{C} \\ a3_{CH4} & a3_{H2O} & a3_{CO} & a3_{H2} & a3_{CO2} & a3_{C} \\ a4_{CH4} & a4_{H2O} & a4_{CO} & a4_{H2} & a4_{CO2} & a4_{C} \\ a5_{CH4} & a5_{H2O} & a5_{CO} & a5_{H2} & a5_{CO2} & a5_{C} \\ a6_{CH4} & a6_{H2O} & a6_{CO} & a6_{H2} & a6_{CO2} & a6_{C} \end{bmatrix}$$

Now, reading in the coefficients

```
> L:=readdata("prothero.dat",6):
  U:=convert(L,matrix):
```

and then including the coefficient vector for carbon (graphite) into the stoichiometric matrix α

```
> c:=stoich(eqs,[C]);
```

$$c := \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

```
> alpha:=stackmatrix(alpha,c);
```

$$\alpha := \begin{bmatrix} -1 & 0 & -1 \\ -1 & -1 & 0 \\ 1 & -1 & 0 \\ 3 & 1 & 2 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

allows the coefficients A_j, $j = 0..6$, to be calculated for the three reactions by the following matrix multiplication:

```
> V:=evalm(U &* alpha);
```

$$V := \begin{bmatrix} 10.454111 & -5.293134 & 2.846880 \\ 20.387169 & 33.694171 & 36.521534 \\ -85.877036 & -57.811586 & -98.41227 \\ 100.248370 & 51.328750 & 103.64693 \\ -57.177818 & -25.508517 & -55.793416 \\ 16.3933072 & 6.7305072 & 15.302810 \\ -1.89006824 & -.73398334 & -1.6986632 \end{bmatrix}$$

The integration constants $\_C1$ and $\_C2$ can be calculated by using special values of $H_P(T)$ and $lnK_P(T)$ under standard conditions ($T = 298.15\,K, P = 1\,bar$) for the equations (1-3) [5]. Inserting the value of the gas constant R and taking into account the conversion factor F between the unit cal (used in Prothero's thermochemical table [4]) and the SI unit joule we eventually arrive at:

```
> R:=8.3143:
  H0[1]:=206185: H0[2]:=-41165: H0[3]:= 74873:
  lnK0[1]:=-57.3621: lnK0[2]:=11.546:
  lnK0[3]:=-20.47458665: F:=4.1868: s:=10^(-3):

> for i from 1 to 3 do
     for k from 0 to 6 do
        A[k]:=F*V[k+1, i]
     od;
     solve({H0[i]=eval(HPT,T=298.15)},_C1);
     assign(%);
     solve({lnK0[i]=eval(lnKPT,T=298.15)},_C2);
     assign(%);
     ln(K[i])=eval(lnKPT);
     print(%); assign(%);
    _C1:='_C1':_C2:='_C2':
  od:
```

$$\ln(K_1) = -.2266129470\,10^{-19}\,T^6 + .2751704836\,10^{-15}\,T^5 - .1439640670\,10^{-11}\,T^4 + .4206807102\,10^{-8}\,T^3 - .7207461325\,10^{-5}\,T^2$$

$$+ .005133144050\,T - 23067.92495\,\frac{1}{T} + 5.264336374\,\ln(T) - 10.97646893$$

$$\ln(K_2) = -.8800218119\,10^{-20}\,T^6 + .1129751855\,10^{-15}\,T^5$$
$$- .6422612785\,10^{-12}\,T^4 + .2153951730\,10^{-8}\,T^3 - .4851992917\,10^{-5}\,T^2$$
$$+ .008483621895\,T + 4698.749277\,\frac{1}{T} - 2.665443083\,\ln(T) + 8.822583615$$

$$\ln(K_3) = -.2036641141\,10^{-19}\,T^6 + .2568659013\,10^{-15}\,T^5$$
$$- .1404783771\,10^{-11}\,T^4 + .4349423747\,10^{-8}\,T^3 - .8259514568\,10^{-5}\,T^2$$
$$+ .009195504045\,T - 8107.550186\,\frac{1}{T} + 1.433592386\,\ln(T) - 3.561949850$$

6 Temperature dependence of the product gas composition

In this section we use the derived mathematical model and analyse the temperature dependence of the gas composition for steam reforming of hexane. Figures 2-4 display the evolution of the reaction extents and the mole fractions over the temperature interval $T = 750..1200\,K$ for two different steam to carbon ratios $N = n_2/n_1$ (below and above the carbon boundary). The nonlinear equation system is solved at a step size of 10 degrees, i.e. 46 times, within an accuray of 4 digits. For this task we used a modified version of the newton procedure from Ross Taylor (Maple share library), which is somewhat more flexible than Maple's built-in solver fsolve, e.g. the Newton step may be limited to some maximum value. First we study the temperature dependence of the reaction system below the carbon boundary with a molar steam to carbon ratio of $N = 1$ and a total pressure of $P = 2\,bar$ (Fig. 2 and 3). The three reaction extents show the expected progression. The strong endothermic methane-steam reaction becomes more and more dominant as the reaction temperature is increased. Simultaneously the extent of the carbon formation reaction (3) and the exothermic water-gas shift (2) decline slowly after a local maximum and at $T = 1200\,K$ reach values indicating that there is theoretically no carbon and only very small carbon dioxide formation in the reactor. Fig. 3 shows the temperature dependence of the corresponding mole fractions for the gas components. At high temperatures only hydrogen and carbon monoxide remain in the gas mixture whereas at low temperatures methane is formed in the reactor. The same qualitative reaction behaviour was found when the steam to carbon ratio increases from $N = 1$ to $N = 2$ (Fig. 4). Under these conditions the reaction extent of the carbon formation becomes negative and the reaction system is above the carbon boundary. Noticeable differences are only found in the values for carbon dioxide which are nearly constant over the temperature range, since the higher water rate in the reaction mixture moves the water-gas shift equilibrium (2) to the right side.

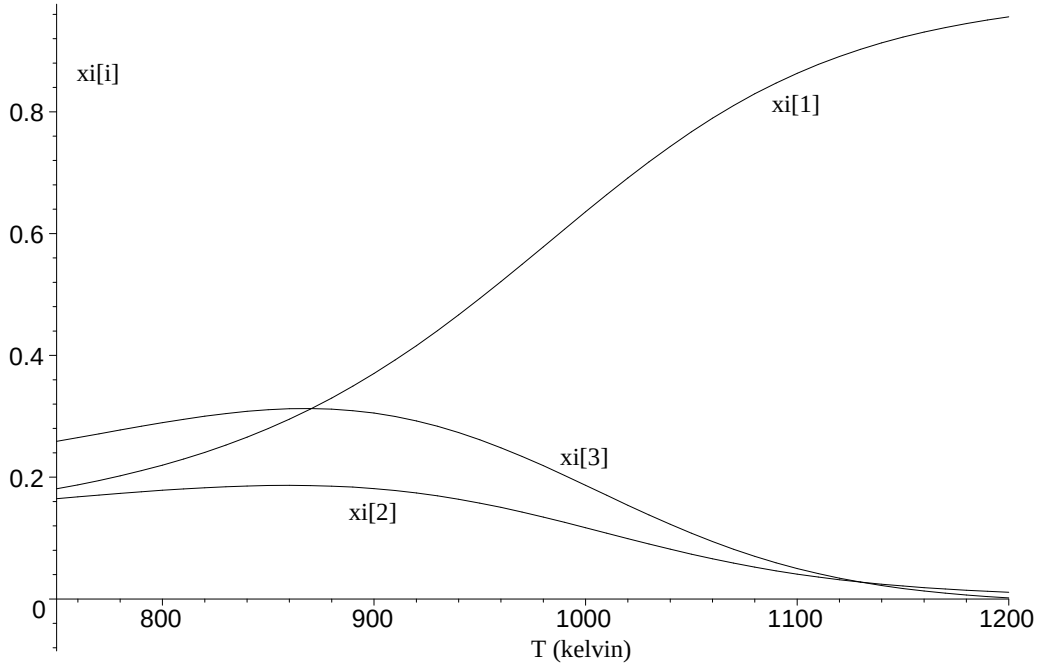

Fig. 2. Temperature dependence of the reaction extents below the carbon boundary for steam reforming of hexane ($P = 2, N = 1, \eta = 14/6$).

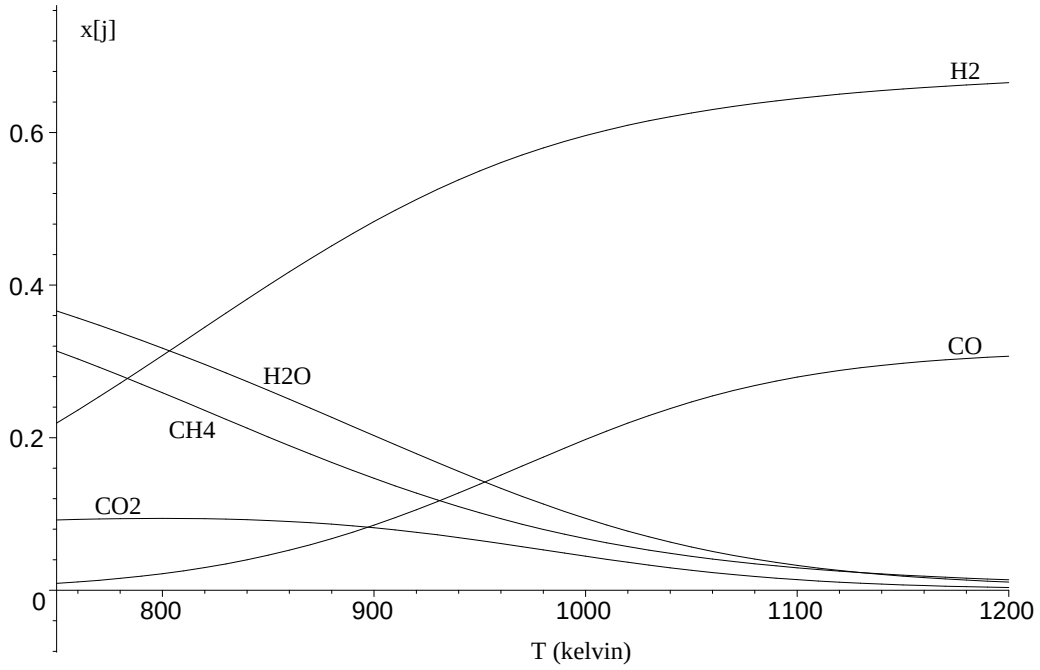

Fig. 3. Temperature dependence of the mole fractions x_j, $j = 1..5$, below the carbon boundary. Same conditions as in Fig. 2.

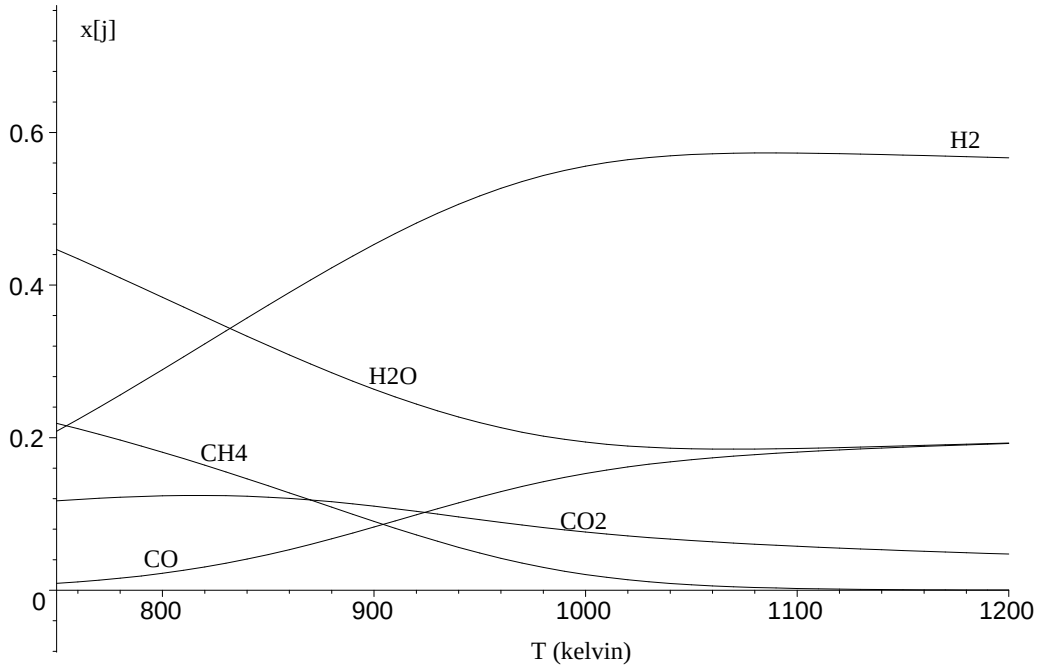

Fig. 4. Temperature dependence of the mole fractions x_j, $j = 1..5$, above the carbon boundary for steam reforming of hexane ($P = 2, N = 2, \eta = 14/6$).

7 Numerical code generation

For more complex parameter studies (T-P diagrams or T-N diagrams) we used the compiler language Fortran to speed up the numerical solution of the nonlinear equation system. A preprocessor gen_fort for Fortran code generation and optimization was implemented in Macrofort [6] (Maple share library), a macro language for Fortran code generation in Maple (see Appendix). Macrofort allows the generation of complete Fortran programs, including declarations and control structures. The Jacobian matrix needed for the Newton algorithm is calculated in exact form by Maple using algorithmic differentiation. We used the JACOBIAN program in the new codegen package in Maple V Release 5 [2]. The numerical solution of the system of linear equations encountered in the Newton iteration is done by applying the LU decomposition routine dgetrs and the general linear equation solver dgetrf from LAPACK [7]. The preprocessor gen_fort is designed to construct complete and ready to compile Fortran code for a given problem size, i.e. larger systems of reaction equations can be studied in the same way as demonstrated here. The following Maple commands show how the preprocessor was used. The left-hand side of the multiparameter equations is passed to the gen_fort program, the Fortran program simprog.f is generated and then compiled using the system command in Maple; the solver linsolve for the linear equations is linked to a resulting executable.

```
> with(share):
  with(macrofor):
  read(gen_fort);
  g:=map(y -> lhs(y),f):
  gen_fort(g,xi,3,x,5,"simprog.f");
  system("f77 -O -o simprog linsolve.o simprog.f");
```

We observed that the new JACOBIAN procedure in the codegen package generates a much more compact Jacobian matrix than the corresonding function jacobian in the linalg package. This can be seen convincingly by applying the cost function to the generated matrices. The command cost, also a function in the codegen package, computes an operation count for the numerical evaluation of a given expression.

```
> with(codegen):
  jacobian([seq(g[i],i=1..3)],[seq(xi[i],i=1..3)]):
  cost(convert(%,listlist));
```

$$362\ additions + 319\ multiplications + 78\ divisions$$

```
> eval(JACOBIAN([seq(unapply(g[i],seq(xi[i],i=1..3)),i=1..3)])
  (seq(xi[i],i=1..3))):
  cost(convert(%,listlist));
```

$$93\ additions + 48\ multiplications + 26\ divisions$$

8 Multiparameter studies

The numerical programs were used for studying the reaction behaviour of steam reforming by varying two reaction parameters simultaneously. First we analyse the temperature ($T = 750..1200\,K$) and the pressure dependence ($P = 1..14\,bar$) for methane reforming with a molar steam to carbon ratio $N = 1$. Fig. 5 shows that the temperature interval where carbon formation occurs becomes smaller when the pressure is increased and that for $P > 12\,bar$ the carbon formation disappears. The plane at $\xi_3 = 0$ separates the reaction conditions where the reaction system is inside (positive values of ξ_3) or outside (negative values of ξ_3) the carbon formation region. The resulting section curve at $\xi_3(P,T) = 0$ marks the carbon boundary for this reaction mixture and is plotted separately as a 2D graph in Fig. 6. The corresponding surface plots of the mole fractions for the various gas components illustrate that the reaction pressure has less influence on the product composition than the temperature (Fig. 7). In contrast to the pressure, the steam to carbon ratio has a strong influence on the product gas composition. Fig. 8 shows the resulting

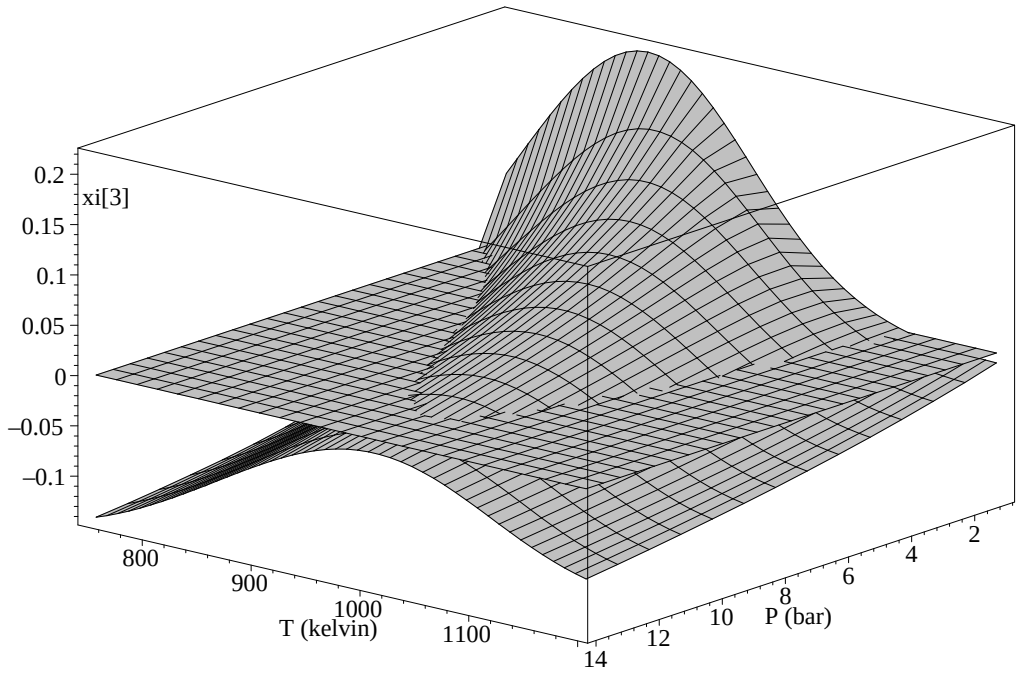

Fig. 5. Temperature and pressure dependence of the reaction extent ξ_3 for steam reforming of methane ($N = 1, \eta = 4$). The carbon boundary is defined by $\xi_3 = 0$

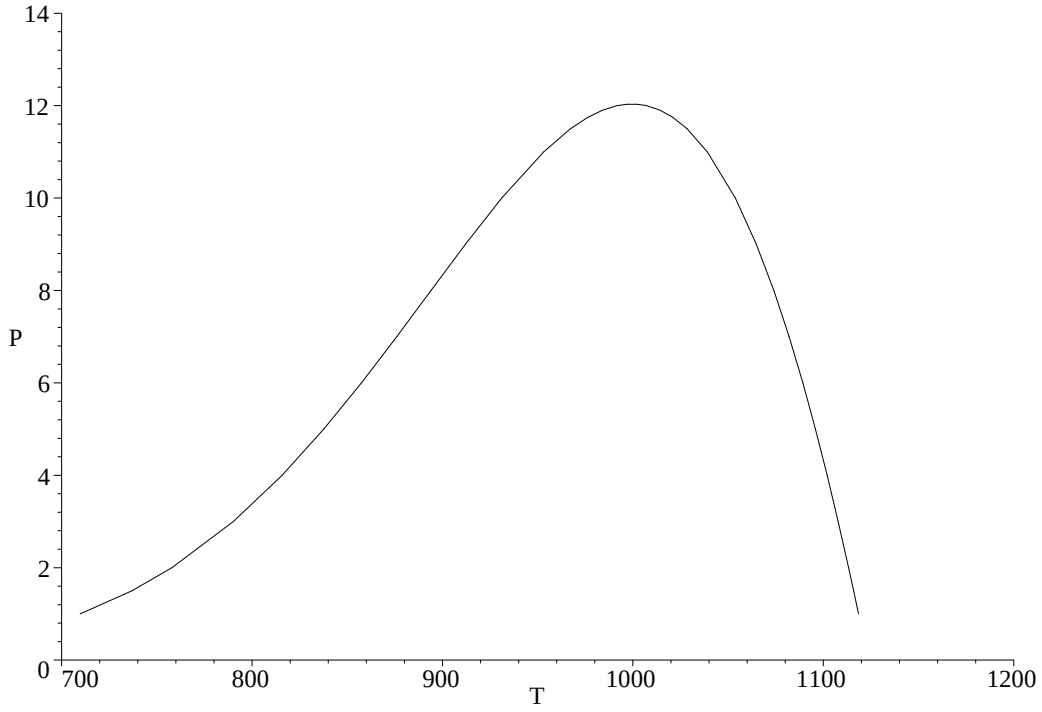

Fig. 6. The resulting carbon boundary $\xi_3(P, T) = 0$ of methane reforming at a steam to carbon ratio $N = 1$.

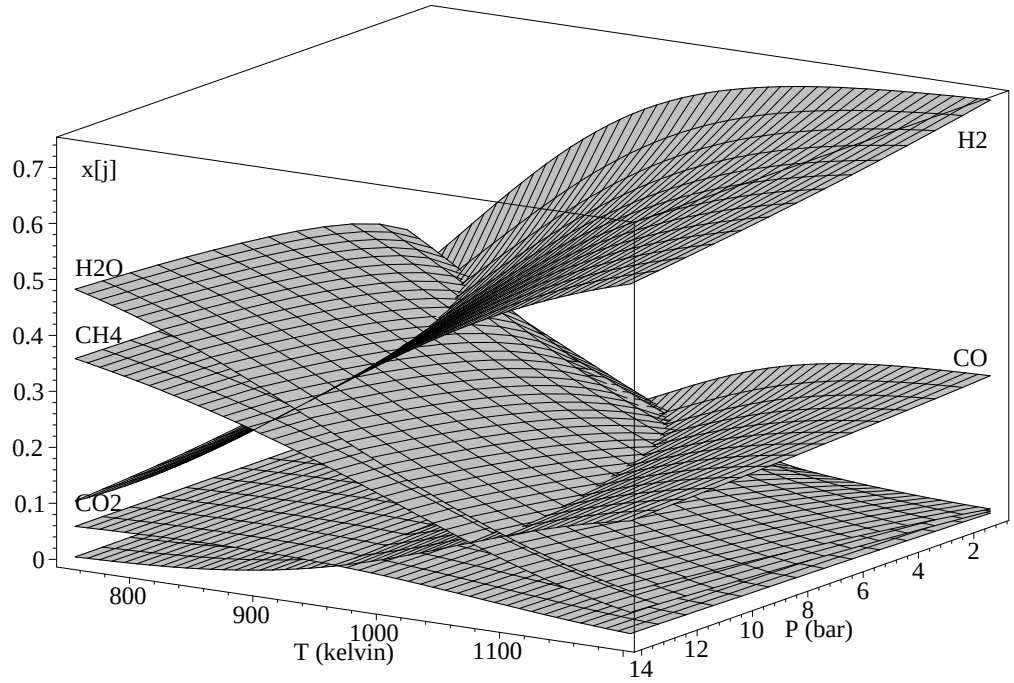

Fig. 7. Temperature and pressure dependence of the mole fractions x_j, $j = 1..5$, for steam reforming of methane. Same conditions as in Fig. 5.

plot by varying the temperature in the same inteval and the steam to carbon ratio over the interval $N = 1.5 - 3.5$ at the constant pressure $P = 2\,bar$. A small increase of the water fraction in the reaction mixture changes the carbon monoxide to carbon dioxide ratio dramatically.

9 Visualization

Numerical data produced by the program simprog are written to a file which is read back into Maple by readdata for visualization.

```
> l:=readdata("result.dat",10):
```

The data in the file "result.dat" consist of floats arranged in 10 columns representing the variables $P, T, x_1, ..., x_5, \xi_1, \xi_2, \xi_3$. The output of the command readdata is a list of rows of data corresponding to the rows of data in the file. As an example we display the Maple code for generating Fig. 5.

```
> ll:=[seq([seq([op(l[n*14+m][1..2]),l[n*14+m][10]],m=1..14)],
       n=0..45)]:
  with(plots):
  xi3:=surfdata(ll,color=grey):
```

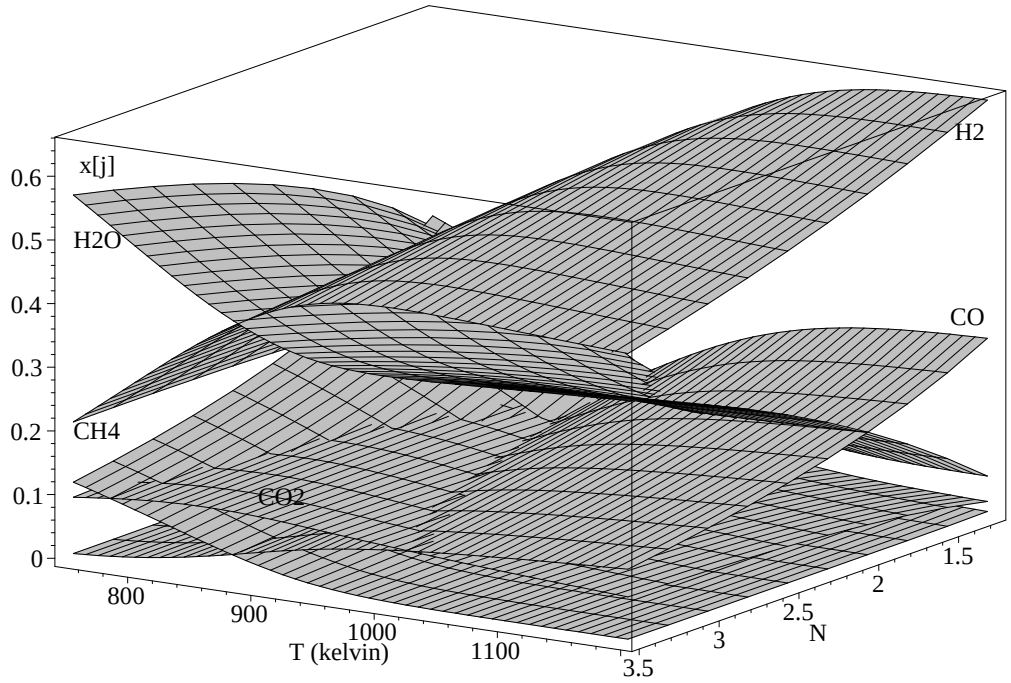

Fig. 8. Temperature and N (steam to carbon ratio, $H_2O : C$) dependence of the mole fractions x_j, $j = 1..5$, for steam reforming of hexane ($P = 2, \eta = 14/6$).

```
xi0:=plot3d(0,1..14,750..1200,color=grey):
t:=textplot3d([14,760,0.18,"xi[3]"]):
display({xi3,xi0,t},orientation=[40,64],axes=boxed,
labels=["P (bar)","T (Kelvin)",""]);
```

For plotting the reaction extent ξ_3 as a function of P and T we used the command surfdata of the plot package plots. First the appropriate grid data structure for surfdata is constructed. Specifically, a surface is input as a list of lists of points where each point is a list of three constants $[x, y, z]$, in our case $[P, T, \xi_3]$.

10 Experimental steam reformer

The thermodynamic model was used to determine the mass and energy balance of an experimental steam reformer for converting organic wastes into hydrogen-rich product gases. Fig. 9 shows the apparatus which was designed for reactor pressures of less than 10 bar and for a maximum carbon mass flow of 250 g(C)/h. The required steam was produced in a commercial steam generator (A), superheated in a heat exchanger (C) up to $350° C$ and fed in a two component noozle where the steam was merged with the liquid hydrocar-

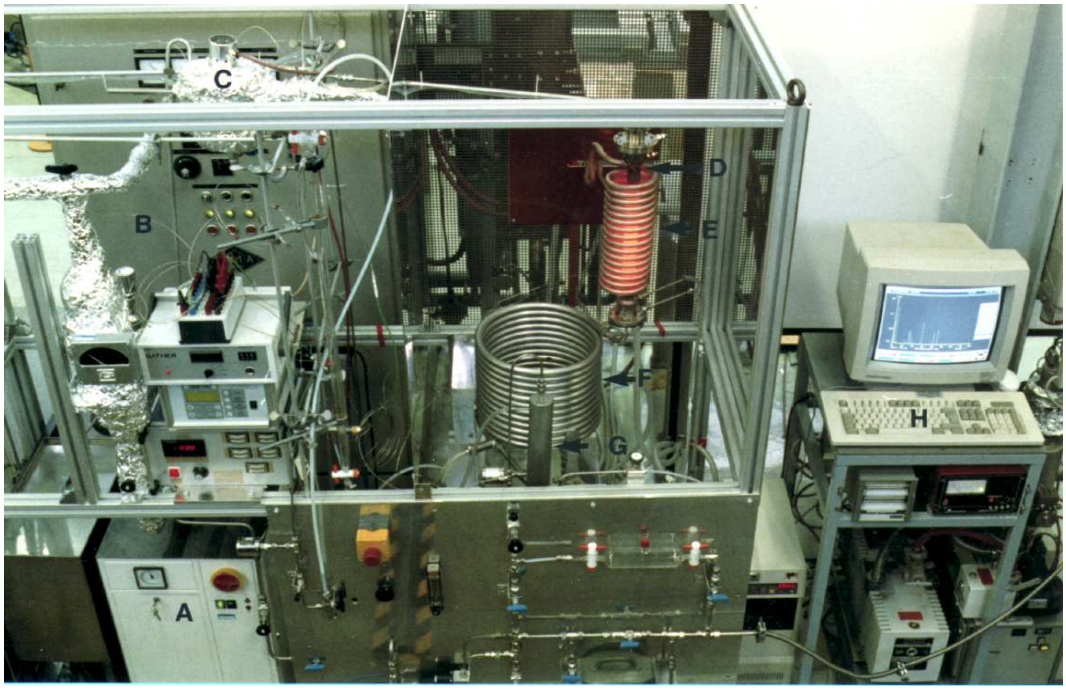

Fig. 9. Experimental steam reformer: (A) steam generator, (B) induction generator, (C) steam preheater, (D) reactor, (E) inductor, (F) coaxial heat exchanger, (G) condensate separator, (H) on-line gas analyser (Quadrupole-MS)

bon which was pumped in from a storage vessel at a constant flow rate. The reaction mixture was sprayed onto the catalyst in the reactor (D) which was heated by an induction furnace (induction generator B, inductor E). The resulting product gas was cooled in a coaxial heat exchanger (F), passed through a condensate separator (G) to remove the excess reaction water. It then flowed into an exhaust gas system, where the gas composition was measured on-line with a calibrated quadrupole mass spectrometer (H). The steam reformer was tested with hexane by measuring the product gas composition while raising the temperature from $600° C$ to $1000° C$ with a high feed charge of the nickel catalyst under the following conditions:

```
H2O/C-ratio:       N = 2.7
Pressure:          P = 2 bar
Space velocity:    SV = 1.2 g(C)/(g(catalyst)*h)
Contact time:      t = 0.34 s
```

Fig. 10 illustrates the resulting data in comparison to the theoretical data obtained under the same reaction conditions by the computational model described above. The temperature dependence of the experimental and the theoretical curves shows the same tendency for all gas components but the absolute values are different. The measured hydrogen gas fractions are approximately 3 vol% too low and the carbon monoxide and carbon dioxide values are a bit too high. This may be due to the remaining fraction of methane in the

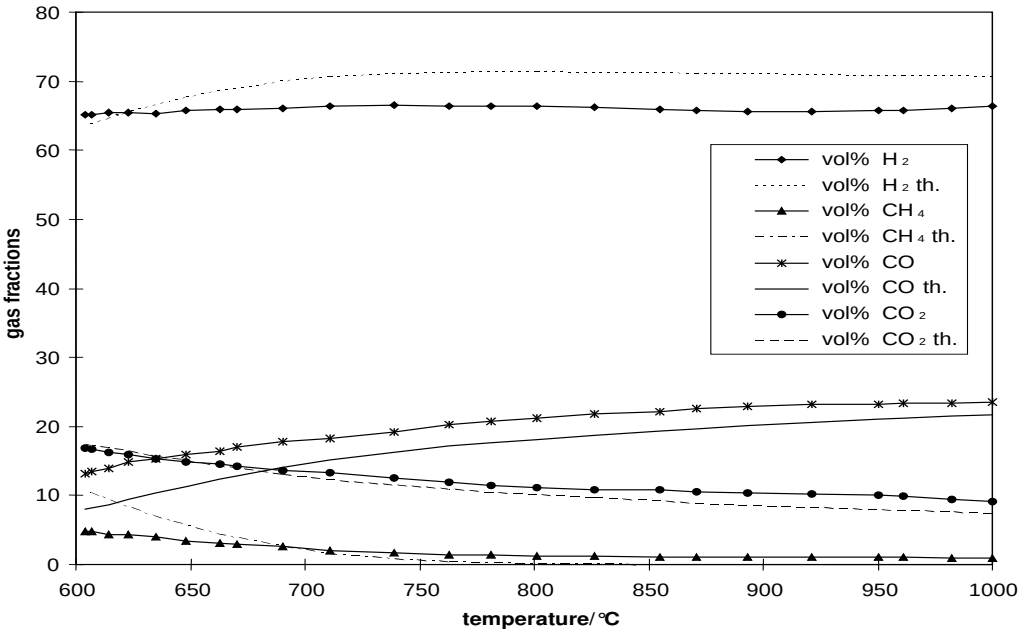

Fig. 10. Temperature dependence of measured and calculated data of the gas composition for steam reforming of hexane ($N = 2.7, P = 2\,bar, \eta = 14/6$).

product gas which is produced by the methane-steam equality (1) and should theoretically be lower at high temperatures.

11 Conclusion

In this article we analysed the thermodynamics of steam reforming by using computing techniques from symbolics, numerics and graphics. It is shown that the integration of the various computing methods normally used separately in scientific problem-solving allows the development of efficient and flexible tools for computer-aided modelling and simulation. The generation of the mathematical model is done automatically by symbolic methods. Then the product gas composition at equilibrium is calculated by solving a multiparameter nonlinear equation system with adapted numerical techniques. For simple calculations we used the numerical functions implemented in Maple (e.g. fsolve) which are quickly applied but which have a relatively slow performance (interpreted). For more complex parameter studies, i.e. for numerically intensive computations, we used the compiler language Fortran. The numerical programs are generated by a flexible preprocessor designed to construct complete and ready to compile Fortran code for a variable problem size. The graphical

visualization of the numerical data is accomplished by using appropriate plot functions in Maple. Finally, we compared the gas fractions measured in an experimental steam reformer at different temperatures with the corresponding calculated (theoretical) values.

All tools and computing techniques described in this article are accessible within an interactive computing environment (Maple's worksheet interface) and easily changeable for the analysis of other (larger) reaction systems. The Maple worksheets and programs developed for this work are available on request from the corresponding author.

References

[1] R. Taylor, Mathematics and Computers in Simulation 35 (1998) 101-119 and 121-146.

[2] M.B. Monagan, K.O. Geddes, K.M. Heal, G. Labahn, S.M. Vorkoetter, Maple V Programming Guide, Springer, New York, (1998)

[3] A. Ovenston and J.R. Walls, Chem. Engineering Sci. 35 (1980) 627-633.

[4] A. Prothero, Comb. Flame 13(1969) 399-408.

[5] I. Barin, Thermochemical Data of Pure Substances, VCH Verlagsgesellschaft, Weinheim, (1993).

[6] C. Gomez, Macrofort: a Fortran code generator in Maple, Maple Share Library, Version 5.0.

[7] J. Dongarra et al., The LAPACK user's guide, SIAM publications, Philadelphia, (1992).

12 Appendix

A listing of the Fortran code generator gen_fort in Macrofort follows:

```
> # Maple program gen_fort in Macrofort
  gen_fort:=proc(ff,xx,d,x1,dd,pg_name)
  local f,fo,ii,jac,jj,j,k,l,lwhile,m,pg,xi,
        zj,y,ipiv,n,nwhile1,llwhile, forloops,fw,fv,
        eps,maxiter,Tstart,Tstep,nTsteps,Pstart,Pstep,nPsteps,
        P,T,znorm,nz,precision;
  # calculate the jacobian matrix
  # jac:=linalg[jacobian]([seq(ff[ii],ii=1..d)],
         [seq(xx[ii],ii=1..d)]);
    jac:=eval(codegen[JACOBIAN]([seq(unapply(ff[ii],
         seq(xx[ii],ii=1..d)),ii=1..d)])(seq(xx[ii],ii=1..d)));
  # pg: MAPLE list describing the main program
  pg:=[[declaref,‘implicit double precision‘,
       [‘(a-h,o-z)‘]],
       [declaref,‘double precision‘, [‘n‘,‘eta‘, ‘maxstep‘]],
       [declaref,‘dimension‘,
       [f[d],zj[d,d],xi[d],y[dd],n[dd]]],
       [declaref,dimension,[ipiv[d]]],
       [writem,output,[‘’eta, vector n:’‘],[]],
       [readm,input,[‘e14.7‘,‘‘.dd.‘e14.7‘],
       [eta,seq(n[ii],ii=1..dd)]],
       [writem,output,[‘’eps, maxstep, maxiter:’‘],[]],
       [readm,input,[‘2e14.7‘,i6],[eps,maxstep, maxiter]],
       [writem,output,[‘’Pstart,Pstep,nPsteps:’‘],[]],
       [readm,input,[‘2e14.7,i6‘],[Pstart,Pstep,nPsteps]],
       [writem,output,[‘’Tstart,Tstep,nTsteps:’‘],[]],
       [readm,input,[‘2e14.7,i6‘],[Tstart,Tstep,nTsteps]],
       [writem,output,[‘’Inital vector xi:’‘],[]],
       [readm,input,[‘‘.d.‘e14.7‘],
       [seq(xi[ii],ii=1..d)]]
      ];
  # while instruction
  fv:=[];
  for ii from 1 to d do
     pushe([equalf,f[ii],ff[ii]],’fv’)
  od;
  lwhile:=[[matrixm,zj,jac],
           [callf,dgetrf,[d,d,zj,d,ipiv,info]],
           [callf,dgetrs,[‘’N’‘,d,1,zj,d,ipiv,f,d,info]],
           [dom,l,1,d,
```

```
          [[if_then_m ,f[l] > maxstep,[equalf,f[l],maxstep]],
          [if_then_m ,f[l] < -maxstep,[equalf,f[l],-maxstep]],
          [equalf,xi[l],-f[l]+xi[l]]]
          ]];
  for ii from 1 to d do
      pushe([equalf,f[ii],ff[ii]],'lwhile')
  od;
  llwhile:=[whilem,znorm(f,d) >= eps,fv,lwhile,maxiter];
  # translating the mole fractions
  fw:=[];
  for ii from 1 to dd do
      pushe ([equalf,y[ii],x1[ii]],'fw')
  od;
  fw:=[op(fw), [writem,4,[`2(1x,e14.7),1x`,``.dd.`e14.7,1x`,
      ``.d.`e14.7,1x,i6`],[P,T,y,xi,nwhile1]]];
  forloops:=[dom,j,1,nTsteps,
            [[equalf,T,Tstart+(j-1)*Tstep],
            [dom,k,1,nPsteps,[[equalf,P,Pstart+(k-1)*Pstep],
            llwhile,fw]]]];
  pg:=[op(pg),[openm,4,`result.dat`,UNKNOWN,forloops]];
  pg:=[programm,newton,pg];
  # fo: MAPLE list describing the subroutine for
  #     computing the quadratic norm of f
  fo:=[[declaref,`implicit double precision`,[`(a-h,o-z)`]],
      [declaref,`dimension`,[f[m]]],
      [equalf,znorm,0.],
      [dom,j,1,d,[equalf,znorm,znorm+f[j]**2]],
      [equalf,znorm,sqrt(znorm)]];
  fo:=[functionm,`double precision`,znorm,[f,m],fo];
  # FORTRAN code generation
  writeto(pg_name);
  init_genfor();
  precision:=double;
  #optimized:=true;
  interface(quiet=true);
  genfor(pg);
  genfor(fo);
  interface(quiet=false);
  writeto(terminal);
end:
```